



HibernateUtil.java Student.java CriteriaTest.java ×

```
19 Session sn = sf.openSession();
20
21 // Student s=new Student();
22 // s.setsId(106);
23 // s.setsName("Sajed");
24 // s.setAddress("Mumbai");
25 // s.setMarks(71);
26
27 Criteria cq = sn.createCriteria(Student.class);
28
29 //if we want to get data on some condition then
30 //1. get records where sId=102
31 cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 //cq.add(Restrictions.gt("sId", 102));
35
36 List<Student> st = cq.list();
37
38
39
40
41 for(Student s:st) {
42
43     System.out.println("Student Id: "+s.getId());
44     System.out.println("Student Name: "+s.getName());
45     System.out.println("Student Addr: "+s.getAddress());
46     System.out.println("Student Marks: "+s.getMarks());
47     System.out.println("-----");
48 }
49
50 sn.close();
```

Console ×

<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:18:49 PM – 8:18:51

JdbcConnectionAccess

[org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator\$ConnectionProviderJdbcConnectionAccess@63f259c3] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.

Aug 17, 2023 8:18:51 PM org.hibernate.internal.SessionImpl createCriteria

WARN: HHH90000022: Hibernate's legacy org.hibernate.Criteria API is deprecated; use the JPA

javax.persistence.criteria.CriteriaQuery instead

Hibernate:

select

this_.sId as sId1_0_0_,

this_.address as address2_0_0_,

this_.marks as marks3_0_0_,

this_.sName as sName4_0_0_

from

Student this_

where

this_.sId=?

Student Id: 102

Student Name: Yogesh

Student Addr: Vaijapur

Student Marks: 68.0

Aug 17, 2023 8:18:51 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop

INFO: HHH10001008: Cleaning up connection pool

[jdbc:mysql://localhost:3306/criteriadb]

Writable

Smart Insert

46:50:1120



Search



29°

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1.1 Kbps
1.5 Kbps

ENG
IN

8:19 PM
8/17/2023


```
19 Session sn = sf.openSession();
20
21 // Student s=new Student();
22 // s.setsId(106);
23 // s.setsName("Sajed");
24 // s.setAddress("Mumbai");
25 // s.setMarks(71);
26
27 Criteria cq = sn.createCriteria(Student.class);
28
29 //if we want to get data on some condition then
30 //1. get records where sId=102
31 //cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 cq.add(Restrictions.gt("sId", 102));
35
36 List<Student> st = cq.list();
37
38
39
40
41 for(Student s:st) {
42
43     System.out.println("Student Id: "+s.getId());
44     System.out.println("Student Name: "+s.getName());
45     System.out.println("Student Addr: "+s.getAddress());
46     System.out.println("Student Marks: "+s.getMarks());
47     System.out.println("-----");
48 }
49
50 sn.close();
```

```
Console x
<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:
    this_.marks as marks3_0_0_,
    this_.sName as sName4_0_0_
from
    Student this_
where
    this_.sId>?
Student Id: 103
Student Name: Mayur
Student Addr: Nashik
Student Marks: 81.0
-----
Student Id: 104
Student Name: Ram
Student Addr: Mumbai
Student Marks: 65.0
-----
Student Id: 105
Student Name: Ishwar
Student Addr: Pune
Student Marks: 59.0
-----
Student Id: 106
Student Name: Sajed
Student Addr: Mumbai
Student Marks: 71.0
-----
Aug 17, 2023 8:19:24 PM
org.hibernate.engine.jdbc.connections.internal.DriverM
anagerConnectionProviderImpl stop
INFO: HHH10001008: Cleaning up connection pool
[jdbc:mysql://localhost:3306/criteriadb]
```



```
19 Session sn = sf.openSession();
20
21 // Student s=new Student();
22 // s.setsId(106);
23 // s.setsName("Sajed");
24 // s.setAddress("Mumbai");
25 // s.setMarks(71);
26
27 Criteria cq = sn.createCriteria(Student.class);
28
29 //if we want to get data on some condition then
30 //1. get records where sId=102
31 //cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 // cq.add(Restrictions.gt("sId", 102));
35
36 //3. get records where sId<104
37 cq.add(Restrictions.lt("sId", 104));
38
39 List<Student> st = cq.list();
40
41
42
43
44 for(Student s:st) {
45
46     System.out.println("Student Id: "+s.getId());
47     System.out.println("Student Name: "+s.getName());
48     System.out.println("Student Addr: "+s.getAddress());
49     System.out.println("Student Marks: "+s.getMarks());
50     System.out.println("-----");
```

Console ×

<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:20:51 PM)

javax.persistence.criteria.CriteriaQuery instead

Hibernate:

```
select
    this_.sId as sId1_0_0_,
    this_.address as address2_0_0_,
    this_.marks as marks3_0_0_,
    this_.sName as sName4_0_0_
from
    Student this_
where
    this_.sId<?
```

Student Id: 101

Student Name: Laxman

Student Addr: Pune

Student Marks: 75.0

Student Id: 102

Student Name: Yogesh

Student Addr: Vaijapur

Student Marks: 68.0

Student Id: 103

Student Name: Mayur

Student Addr: Nashik

Student Marks: 81.0

Aug 17, 2023 8:20:51 PM

```
org.hibernate.engine.jdbc.connections.internal.DriverM
anagerConnectionProviderImpl stop
```

INFO: HHH10001008: Cleaning up connection pool

[jdbc:mysql://localhost:3306/criteriadb]

Writable

Smart Insert

37:42:910



Search



29°

0.0 Kbps
0.0 KbpsENG
IN8:20 PM
8/17/2023


```
19 Session sn = sf.openSession();
20
21 // Student s=new Student();
22 // s.setsId(106);
23 // s.setsName("Sajed");
24 // s.setAddress("Mumbai");
25 // s.setMarks(71);
26
27 Criteria cq = sn.createCriteria(Student.class);
28
29 //if we want to get data on some condition then
30 //1. get records where sId=102
31 //cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 // cq.add(Restrictions.gt("sId", 102));
35
36 //3. get records where sId<104
37 //cq.add(Restrictions.lt("sId", 104));
38
39 //4. get records where student marks is between 65 and 80
40 cq.add(Restrictions.between("marks", 65.0, 80.0));
41
42 List<Student> st = cq.list();
43
44
45
46
47 for(Student s:st) {
48
49     System.out.println("Student Id: "+s.getId());
50     System.out.println("Student Name: "+s.getName());
```

Console ×

<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:23:50)

```
        this_.marks as marks3_0_0_,
        this_.sName as sName4_0_0_
    from
        Student this_
    where
        this_.marks between ? and ?
```

```
Student Id: 101
Student Name: Laxman
Student Addr: Pune
Student Marks: 75.0
```

```
-----
Student Id: 102
Student Name: Yogesh
Student Addr: Vaijapur
Student Marks: 68.0
```

```
-----
Student Id: 104
Student Name: Ram
Student Addr: Mumbai
Student Marks: 65.0
```

```
-----
Student Id: 106
Student Name: Sajed
Student Addr: Mumbai
Student Marks: 71.0
```

Aug 17, 2023 8:23:51 PM

```
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop
INFO: HHH10001008: Cleaning up connection pool
[jdbc:mysql://localhost:3306/criteriadb]
```

3

Writable

Smart Insert

40:56:1031


```

19 Session sn = sf.openSession();
20
21 // Student s=new Student();
22 // s.setsId(106);
23 // s.setsName("Sajed");
24 // s.setAddress("Mumbai");
25 // s.setMarks(71);
26
27 Criteria cq = sn.createCriteria(Student.class);
28
29 //if we want to get data on some condition then
30 //1. get records where sId=102
31 //cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 // cq.add(Restrictions.gt("sId", 102));
35
36 //3. get records where sId<104
37 //cq.add(Restrictions.lt("sId", 104));
38
39 //4. get records where student marks is between 65 and 80
40 //cq.add(Restrictions.between("marks", 65.0, 80.0));
41
42 //5. get records where address start with 'M'
43 cq.add(Restrictions.like("address", "M%"));
44
45
46 List<Student> st = cq.list();
47
48
49
50

```

Console ×

<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:26:24)

Connection will be set into auto-commit mode.

Aug 17, 2023 8:26:25 PM

org.hibernate.internal.SessionImpl createCriteria

WARN: HHH90000022: Hibernate's legacy

org.hibernate.Criteria API is deprecated; use the JPA

javax.persistence.criteria.CriteriaQuery instead

Hibernate:

select

this_.sId as sId1_0_0_,

this_.address as address2_0_0_,

this_.marks as marks3_0_0_,

this_.sName as sName4_0_0_

from

Student this_

where

this_.address like ?

Student Id: 104

Student Name: Ram

Student Addr: Mumbai

Student Marks: 65.0

Student Id: 106

Student Name: Sajed

Student Addr: Mumbai

Student Marks: 71.0

Aug 17, 2023 8:26:25 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop

INFO: HHH10001008: Cleaning up connection pool

[jdbc:mysql://localhost:3306/criteriadb]

3

Writable

Smart Insert

43 : 52 : 1136



Search



29°

1.0 Kbps
0.3 KbpsENG
IN8:26 PM
8/17/2023


```
19 Session sn = sf.openSession();
20
21 // Student s=new Student();
22 // s.setsId(106);
23 // s.setsName("Sajed");
24 // s.setAddress("Mumbai");
25 // s.setMarks(71);
26
27 Criteria cq = sn.createCriteria(Student.class);
28
29 //if we want to get data on some condition then
30 //1. get records where sId=102
31 //cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 // cq.add(Restrictions.gt("sId", 102));
35
36 //3. get records where sId<104
37 //cq.add(Restrictions.lt("sId", 104));
38
39 //4. get records where student marks is between 65 and 80
40 //cq.add(Restrictions.between("marks", 65.0, 80.0));
41
42 //5. get records where address start with 'M'
43 //cq.add(Restrictions.like("address", "M%"));
44
45 //6. get records where address end with 'e'
46 cq.add(Restrictions.like("address", "%e"));
47
48
49 List<Student> st = cq.list();
50
```

Console ×

<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:27:33 PM)

Connection will be set into auto-commit mode.

Aug 17, 2023 8:27:33 PM

org.hibernate.internal.SessionImpl createCriteria

WARN: HHH90000022: Hibernate's legacy

org.hibernate.Criteria API is deprecated; use the JPA

javax.persistence.criteria.CriteriaQuery instead

Hibernate:

select

this_.sId as sId1_0_0_,

this_.address as address2_0_0_,

this_.marks as marks3_0_0_,

this_.sName as sName4_0_0_

from

Student this_

where

this_.address like ?

Student Id: 101

Student Name: Laxman

Student Addr: Pune

Student Marks: 75.0

Student Id: 105

Student Name: Ishwar

Student Addr: Pune

Student Marks: 59.0

Aug 17, 2023 8:27:33 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop

INFO: HHH10001008: Cleaning up connection pool

[jdbc:mysql://localhost:3306/criteriadb]

Writable

Smart Insert

43:11:1095



Search



29°

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

0.0 Kbps
1.4 KbpsENG
IN8:27 PM
8/17/2023


```
22 //      s.setsId(106);
23 //      s.setsName("Sajed");
24 //      s.setAddress("Mumbai");
25 //      s.setMarks(71);
26
27 Criteria cq = sn.createCriteria(Student.class);
28
29 //if we want to get data on some condition then
30 //1. get records where sId=102
31 //cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 //cq.add(Restrictions.gt("sId", 102));
35
36 //3. get records where sId<104
37 //cq.add(Restrictions.lt("sId", 104));
38
39 //4. get records where student marks is between 65 and 80
40 //cq.add(Restrictions.between("marks", 65.0, 80.0));
41
42 //5. get records where address start with 'M'
43 //cq.add(Restrictions.like("address", "M%"));
44
45 //6. get records where address end with 'e'
46 //cq.add(Restrictions.like("address", "%e"));
47
48 //7.get records where sName contains 'e' character
49 cq.add(Restrictions.like("sName", "%e%"));
50
51
52 List<Student> st = cq.list();
53
```

Console ×

<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:28:46 PM)

Connection will be set into auto-commit mode.

Aug 17, 2023 8:28:47 PM

org.hibernate.internal.SessionImpl createCriteria

WARN: HHH90000022: Hibernate's legacy

org.hibernate.Criteria API is deprecated; use the JPA

javax.persistence.criteria.CriteriaQuery instead

Hibernate:

select

```
    this_.sId as sId1_0_0_,
    this_.address as address2_0_0_,
    this_.marks as marks3_0_0_,
    this_.sName as sName4_0_0_
```

from

Student this_

where

this_.sName like ?

Student Id: 102

Student Name: Yogesh

Student Addr: Vaijapur

Student Marks: 68.0

Student Id: 106

Student Name: Sajed

Student Addr: Mumbai

Student Marks: 71.0

Aug 17, 2023 8:28:47 PM

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop

INFO: HHH10001008: Cleaning up connection pool

[jdbc:mysql://localhost:3306/criteriadb]

Writable

Smart Insert

48 : 59 : 1296



Search



29°

1.3 Kbps
1.3 KbpsENG
IN8:29 PM
8/17/2023


```

31 //cq.add(Restrictions.eq("sId", 102));
32
33 //2. get records where sId>102
34 //cq.add(Restrictions.gt("sId", 102));
35
36 //3. get records where sId<104
37 //cq.add(Restrictions.lt("sId", 104));
38
39 //4. get records where student marks is between 65 and 80
40 //cq.add(Restrictions.between("marks", 65.0, 80.0));
41
42 //5. get records where address start with 'M'
43 //cq.add(Restrictions.like("address", "M%"));
44
45 //6. get records where address end with 'e'
46 //cq.add(Restrictions.like("address", "%e"));
47
48 //7. get records where sName contains 'e' character
49 //cq.add(Restrictions.like("sName", "%e%"));
50
51 //8. get records where sName contains 'M' character
52 cq.add(Restrictions.ilike("sName", "%M%"));
53
54 List<Student> st = cq.list();
55
56
57
58
59 for(Student s:st) {
60
61     System.out.println("Student Id: "+s.getId());
62     System.out.println("Student Name: "+s.getName());

```

Console ×

<terminated> CriteriaTest [Java Application] D:\setup-jdk\jdk1.8.0_211 (1)\jdk1.8.0_211\bin\javaw.exe (Aug 17, 2023, 8:33:58)

javax.persistence.criteria.CriteriaQuery instead

Hibernate:

```

select
    this_.sId as sId1_0_0_,
    this_.address as address2_0_0_,
    this_.marks as marks3_0_0_,
    this_.sName as sName4_0_0_
from
    Student this_
where
    lower(this_.sName) like ?

```

Student Id: 101

Student Name: Laxman

Student Addr: Pune

Student Marks: 75.0

Student Id: 103

Student Name: Mayur

Student Addr: Nashik

Student Marks: 81.0

Student Id: 104

Student Name: Ram

Student Addr: Mumbai

Student Marks: 65.0

Aug 17, 2023 8:34:00 PM

```

org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop

```

INFO: HHH10001008: Cleaning up connection pool

[jdbc:mysql://localhost:3306/criteriadb]

11

Writable

Smart Insert

51:48:1391